Technische Universität Berlin · An-Institut Deutsche Telekom Laboratories

Online Virtual Network Management

Gregor Schaffrath · Stefan Schmid · Anja Feldmann

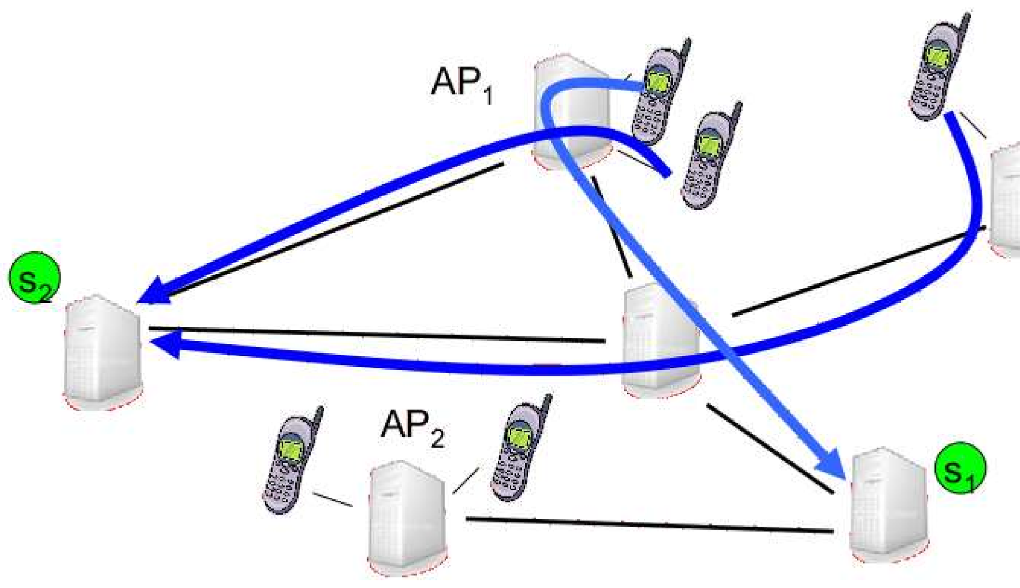
Deutsche Telekom Laboratories
An-Institut der Technischen Universität Berlin

Motivation

Facilitating Management of Virtual Networks

- Economic aspects
  - Dynamic & efficient resource usage
  - New business fields and models
- Security aspects
  - Domain isolation
- Operational aspects
  - Abstraction
  - Out-of-band debugging
  - Potentially higher fault tolerance

Scenario and Goal

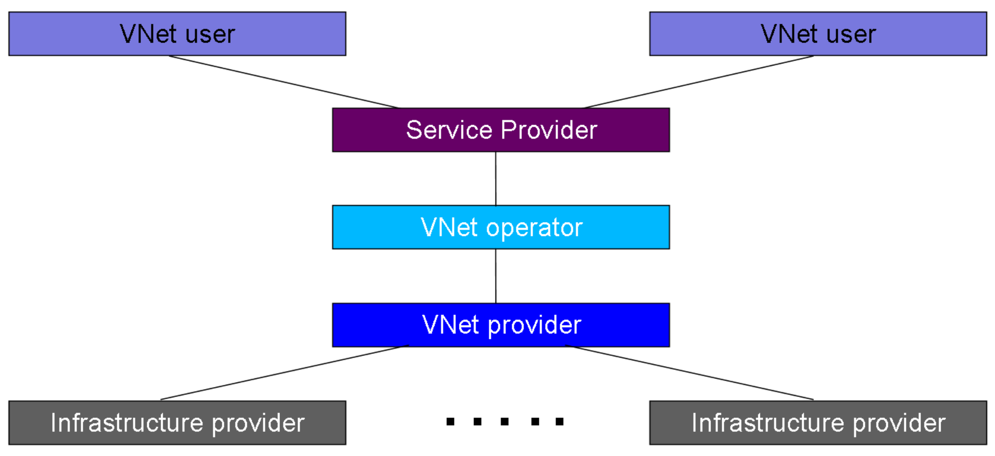


⇒ Effective, economic management of Virtual Networks

Aspects

- Challenge
  - Unpredictable demand
  - Dynamics and flexibility
- Migration protocols
  - Online algorithm
  - Offline algorithm
- Techniques
  - Competitive analysis
  - Dynamic programming

Players



Services

Infrastructure Provider : Provides Virtual Resources and Resource Control Interface

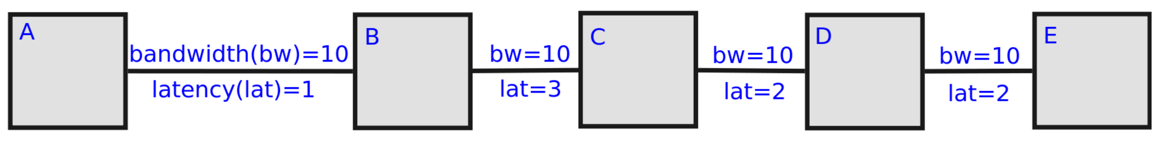
VNet Provider : Assembles Virtual Networks

VNet Operator : Operates, controls, manages virtual networks

Service Provider : Service level customer support

Server Migration

Online MIGration Algorithm

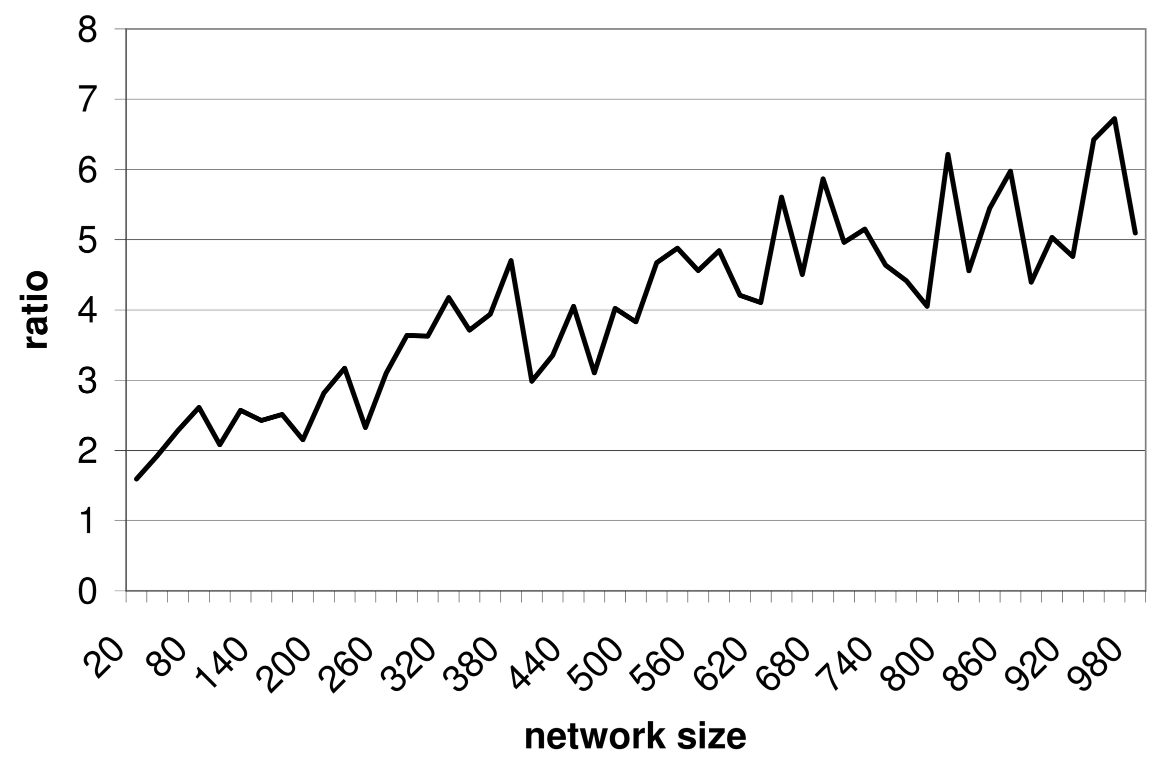


A	B	C	D	E	MIG
(1*1)+(1*6) 7	0+5 5	3+2 5	5+0 5	7+2 9	0+7 7 (A)
7+4+8 19	5+10 15	5+4 9	5+4 9	9+4 13	7+10+4 21 (C)
+10 29	+8 23	+2 11	+2 11	+6 27	+2 23 (C)
0	0	0	0	0	23 (C)

Strike balance between $Cost_{acc}^{MIG}$ and $Cost_{mig}^{MIG}$

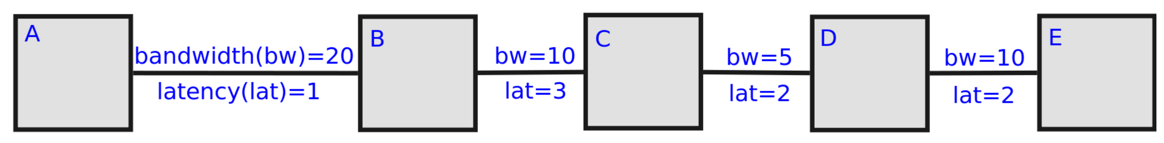
- Let $\beta = max_p\{Cost_{mig}(p,t)\}$
- Count $L_v = \sum_t Cost_{acc}(v,t)\ \forall v \in V$
- When $L_v \geq \beta$ for server location, end phase, and migrate to v' with $L_{v'} < \beta$
- When $L_v \geq \beta\ \forall v \in V$, end epoch ε, and reset $L_v\ \forall v \in V$

Competitiveness



1. Def.(ε), Def.(β) ⇒ $\forall \varepsilon_i : OPT(\varepsilon_i) \geq \beta$
2. H_n migrations expected ⇒ $H_n + 1$ phases expected
3. (2) ⇒ $MIG(\varepsilon_i) \leq \beta H_n + \beta(H_n + 1) = \beta O(\log n)$
4. (1), (3) ⇒ Ratio $\rho \leq \frac{\beta O(\log n)}{\beta} = O(\log n)$

OPTimal Offline Algorithm

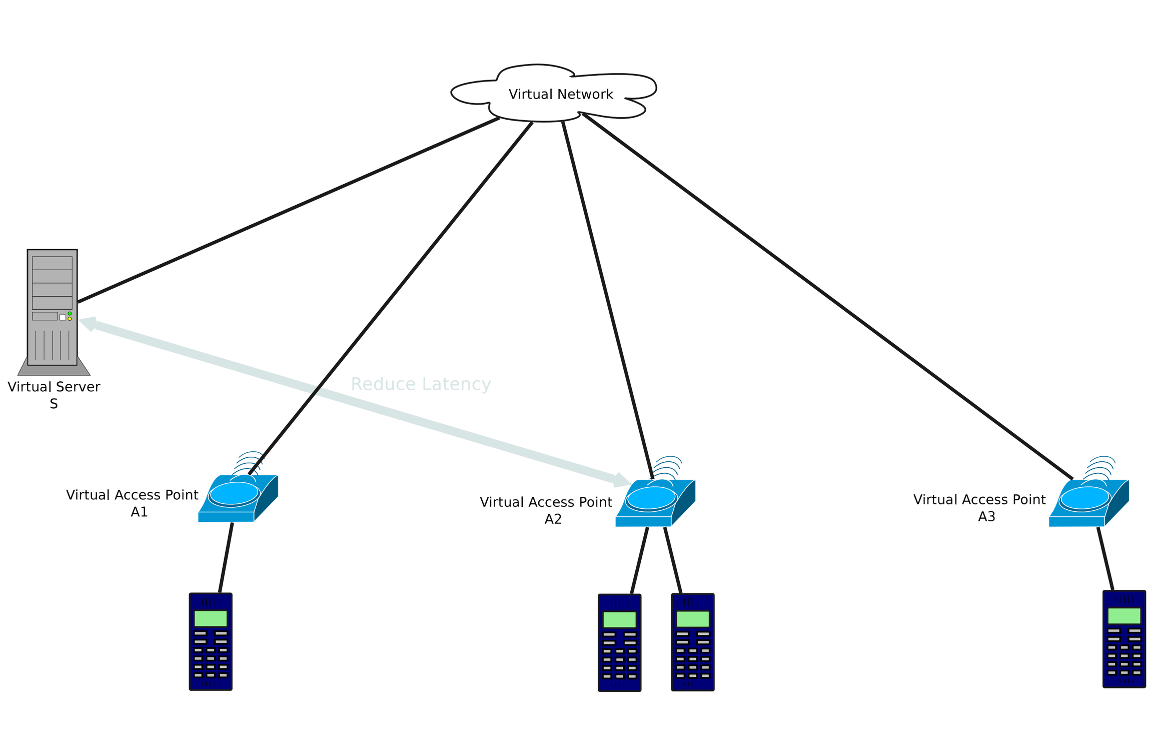


A	B	C	D	E
0+(2*0)+(5*8) 40 (A)	5+37 42 (A)	10+28 38 (A)	20+22 44 (A)	20+16 36 (A)
[40+0|47|48|64|56] +42 82 (A)	[45|42|48|64|56] +35 77 (B)	[50|52|38|64|56] +14 52 (C)	[60|62|58|44|46] +0 44 (D)	[60|62|58|54|36] +14 50 (E)
[90|90|70|72|78] 70 (C)	[92|82|67|69|75] 67 (C)	[112|107|72|84|90] 72 (C)	[136|131|98|78|94] 78 (D)	[150|145|120|102|98] 98 (E)

Dynamic programming

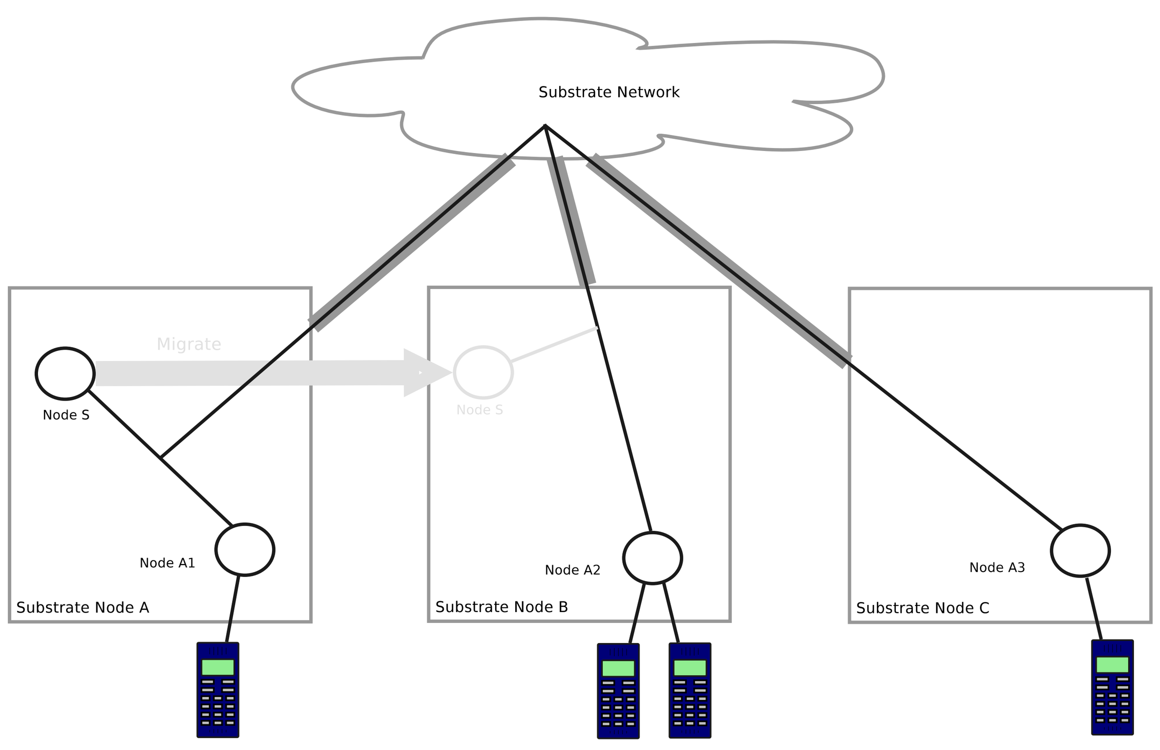
- $opt[t][v]$ matrix with minimal cost
  $opt[0][v] = Cost_{mig}(v_0, v) + \sum_{w \in \sigma_0} Cost_{acc}(w,v)$
  $opt[t][v] =$
  $min_{v, v_{t-1} \in V}(opt[t-1][v_{t-1}] + Cost_{mig}(v_{t-1}, v) + \sum_{w \in \sigma_t} Cost_{acc}(w,v))$
- remember predecessor $v_{t-1} \in V$
- Optimal substructure property

Triggering Migration



VNO view:

- No knowledge of Substrate required
- SP requests latency reduction
- VNO changes virtual resource requirements
- VNO negotiates with VNP

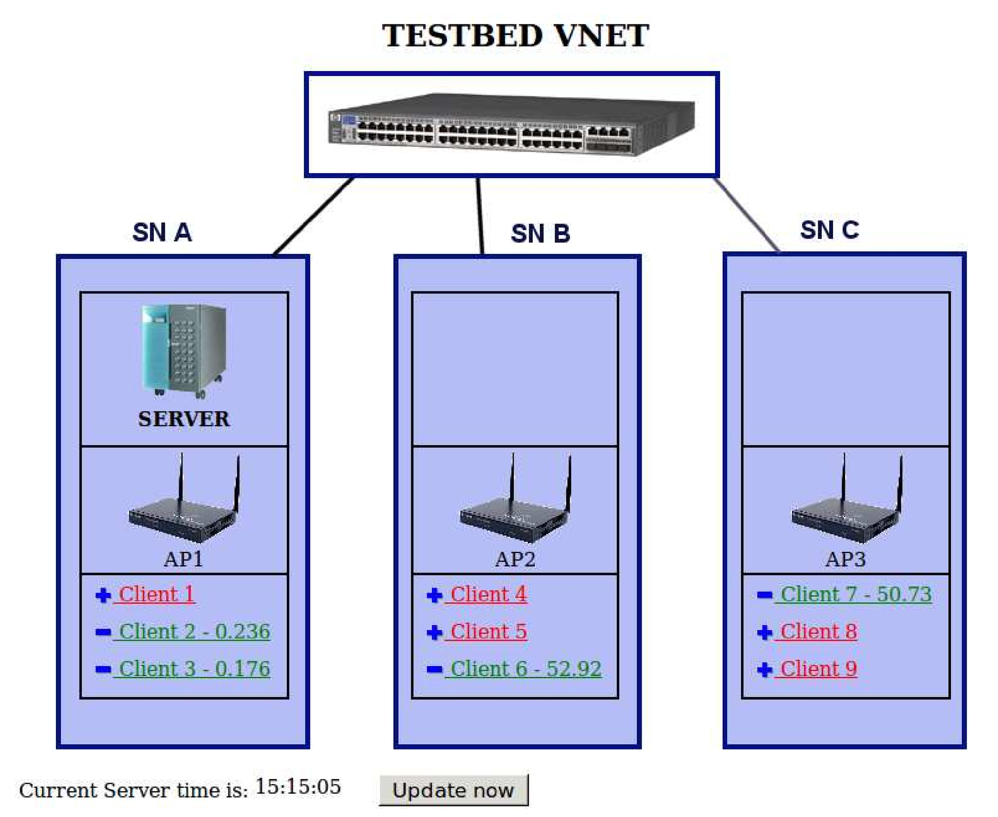


PIP view:

- No knowledge of VNet internal semantics required
- Receives updated requirements
- Initiates migration to effect latency drop

Testbed

TESTBED VNET



- Distributed Virtual Network Testbed
- Proof-of-concept implementations